NASA-CR-195970

ICASE

A TIME-SPLIT DIFFERENCE SCHEME

FOR THE COMPRESSIBLE NAVIER-STOKES EQUATIONS WITH

APPLICATIONS TO FLOWS IN SLOTTED NOZZLES

John Strikwerda

Report No. 80-27

October 20, 1980

(NASA-CR-195970) A TIME-SPLIT DIFFERENCE SCHEME FOR THE COMPRESSIBLE NAVIER-STOKES EQUATIONS WITH APPLICATIONS TO FLOWS IN SLOTTED NOZZLES (ICASE) 30 p

N94-71978

Unclas

Z9/34 0012091

INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING

NASA Langley Research Center, Hampton, Virginia

Operated by the

UNIVERSITIES SPACE USRA RESEARCH ASSOCIATION

A TIME-SPLIT DIFFERENCE SCHEME FOR THE COMPRESSIBLE NAVIER-STOKES EQUATIONS WITH APPLICATIONS TO FLOWS IN SLOTTED NOZZLES

John Strikwerda
Institute for Computer Applications in Science and Engineering

ABSTRACT

A time-split finite difference scheme for the compressible Navier-Stokes equations is presented and discussed. The scheme is suited for use on vector pipeline processors and has been implemented on the Control Data Corporation STAR-100 and CYBER 203 vector processor. The scheme was used to solve for the steady laminar flow in a two-dimensional converging-diverging nozzle with suction slots. The scheme has three splittings, two one-dimensional hyperbolic schemes for the inviscid terms and a parabolic scheme for the viscous terms.

Work was supported under NASA Contract No. NAS1-15810 while the author was in residence at ICASE, NASA Langley Research Center, Hampton, VA 23665.

Table of Contents

1. INTRODUCTION

This paper presents a time-split difference scheme for solving the compressible Navier-Stokes equations and describes the implementation of that scheme on the Control Data Corporation STAR-100 vector processor. Results are presented of computations which use that scheme to compute the laminar flow in converging-diverging nozzles with sunction slots. The scheme is highly vectorizable and suitable for pipeline processors. An interesting feature of the application is that the computational grid is not rectangular but rather two rectangular regions joined along part of one side.

The scheme has been used to compute flows in both two-dimensional and axisymmetric nozzles. The nozzles were either conventional converging-diverging Laval nozzles or nozzles with suction slots located ahead of the throat. The purpose of these computations was to aid in the design of the quiet wind tunnel being developed at NASA Langley Research Center, e.g. Beckwith (1975), Anders, et al. (1977).

This work appears to be the first that uses the full Navier-Stokes equations to solve for the flow in a nozzle at high Reynolds number. In related work, Cline (1976) has used the full Navier-Stokes equations for nozzles at lower Reynolds numbers. Various researchers have employed the slender-channel approximation for nozzles, e.g. Rae (1971) and Mitra and Fiebig (1975); however, the nozzles treated in this paper are not slender enough to be amenable to slender channel approximations. Thomas (1979) uses the Navier-Stokes equations with the parabolic approximation for three-dimensional flows in non-axisymmetric nozzles.

A modification of the method described in this paper has also been used to compute the flow at a wing-elevon junction. This work will be described in a forthcoming report by Walsh and Strikwerda (1980).

Time-split difference schemes have been presented by several authors. Strang (1968) and Gottlieb (1972) have discussed time-split difference schemes for general hyperbolic and parabolic systems and MacCormack (1971) and Abarbanel and Gottlieb (1980) presented time-split schemes for the compressible Navier-Stokes equations. The scheme presented here differs from that of MacCormack in that the viscous terms are split from the inviscid terms. The viscous terms are not split further as advocated by Abarbanel and Gottlieb (1980).

2. THE DIFFERENCE SCHEME

The difference scheme for the Navier-Stokes equations is a time-split scheme with three splittings. One splitting encompasses the parabolic or viscous terms and two splittings are for the hyperbolic or inertial terms, one for each direction. (For the three-dimensional equations a fourth splitting would be added for the inertial terms in the extra dimension.) Each of the corresponding operators is of the predictor-corrector type. The total scheme can be described as a set of three operators applied in sequence so as to be consistent and second-order accurate.

Before describing the difference scheme in detail, we discuss time-splitting in a more general setting, (see also Gottlieb (1972), Strang (1968)). Consider an evolution equation

$$u_t = Au + Bu + Cu \quad , \tag{1.1}$$

where A, B, and C are linear operators of some type. Then given u at time t_0 to compute u at $t_0 + \Delta t$, we may approximate the above equation by

$$u_t = 3A\,u \quad \text{for} \quad t_0 \le t < t_0 + \frac{1}{3}\Delta t,$$

$$u_t = 3B\,u \quad \text{for} \quad t_0 + \frac{1}{3}\Delta t \le t < t_0 + \frac{2}{3}\Delta t,$$

$$u_t = 3C\,u \quad \text{for} \quad t_0 + \frac{2}{3}\Delta t \le t \le t_0 + \Delta t.$$

In this way, the exact solution $u(t_0 + \Delta t)$, which may be written as

$$e^{(A+B+C)\Delta t}u(t_0)\ , \tag{1.2}$$

is approximated by

$$e^{C\Delta t}e^{B\Delta t}e^{A\Delta t}u(t_0). \tag{1.3}$$

Unless the operators A, B, and C commute the expression (1.3) is not equal to the exact solution (1.2), but will be an approximation to within $O(\Delta t^2)$. Strang (1968) has shown that by reversing the order of the splitting in the next time step $u(t_0 + 2\Delta t)$ can be approximated to within $O(\Delta t^3)$ and thus the overall method is second-order accurate in time. When the operators A, B, and C are differential operators and are approximated by difference operators and are approximated by difference operators then the above approach gives rise to a time-split finite difference scheme.

The Navier-Stokes equations for two dimensions may be written as

$$U_t + F_x + G_y - V = 0,$$

where

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}, \qquad F = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E+p)u \end{pmatrix}, \qquad G = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E+p)v \end{pmatrix},$$

and

$$V = \begin{pmatrix} 0 \\ \frac{\partial}{\partial x}\tau_{xx} + \frac{\partial}{\partial y}\tau_{xy} \\ \frac{\partial}{\partial x}\tau_{xy} + \frac{\partial}{\partial y}\tau_{yy} \\ \frac{\partial}{\partial x}(u\tau_{xx} + v\tau_{xy}) + \frac{\partial}{\partial y}(u\tau_{xy} + v\tau_{yy}) + \frac{c_p}{\sigma}\left(\frac{\partial}{\partial x}\left(\mu \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial T}{\partial u}\right)\right. \end{pmatrix},$$

where

$$\tau_{xx} = \frac{4}{3}\mu \frac{\partial u}{\partial x} - \frac{2}{3}\mu \frac{\partial v}{\partial y} \quad ,$$

$$\tau_{xy} = \mu \frac{\partial u}{\partial y} + \mu \frac{\partial v}{\partial x} \quad ,$$

$$\tau_{yy} = \frac{4}{3}\mu \frac{\partial v}{\partial y} - \frac{2}{3}\mu \frac{\partial u}{\partial x} \quad .$$

The equation of state is

$$p = \rho T c_v(\gamma-1) \quad ,$$

and the viscosity is given by Sutherland's law.

The energy E is defined by

$$E = \rho(c_v T + \tfrac{1}{2}(u^2 + v^2)).$$

The gas constant γ is 1.4, the Prandtl number σ is .72, c_v is the coefficient of specific volume. The vectors F and G represent the inertial effects of the flow and V represents the viscous forces. The three splittings used in the scheme correspond to the terms F_x, G_y, and V.

Considering first the viscous terms, the system to be integrated for a step of size $\frac{1}{3}\Delta t$ can be written

$$\begin{aligned}
\tfrac{1}{3}\rho_t &= 0, \\
\tfrac{1}{3}\rho u_t &= \tfrac{4}{3}(\mu u_x)_x + (\mu u_y)_y - \tfrac{2}{3}(\mu v_y)_x + (\mu v_x)_y, \\
\tfrac{1}{3}\rho v_t &= (\mu u_y)_x - \tfrac{2}{3}(\mu u_x)_y + (\mu v_x)_x + \tfrac{4}{3}(\mu v_y)_y, \\
\tfrac{1}{3}\rho c_v T_t &= (kT_x)_x + (kT_y)_y + \Phi,
\end{aligned} \tag{2.1}$$

where

$$\Phi = 2\mu\{u_x^2 + v_y^2 - \tfrac{1}{3}(u_x + v_y)^2\} + \mu(u + v_x)^2,$$

$$k = c_p \mu/\sigma.$$

The subscripts t, x, and y denote differentiation.

We now rewrite the equations in terms of the independent variables (ξ,η). The mapping from the physical (x,y) coordinates to the computational (ξ,η) coordinates will be described in more detail in the appendix, but for now it is sufficient to write it as

$$\xi = \xi(x) \qquad \eta = \eta(x,y).$$

In the transformed coordinates the equation for u is

$$\begin{aligned}\frac{1}{3}\rho u_t = {} & \frac{4}{3}\xi_x(\mu\xi_x u_\xi)_\xi + \frac{4}{3}\eta_x(\mu\xi_x u_\xi)_\eta \\ & + \frac{4}{3}\xi_x(\mu\eta_x u_\eta)_\xi + \frac{4}{3}\eta_x(\mu\eta_x u_\eta)_\eta \\ & + \eta_y(\mu\eta_y u_\eta)_\eta \\ & - \frac{2}{3}\xi_x(\mu\eta_y v_\eta)_\eta - \frac{2}{3}\eta_x(\mu\eta_y v_\eta)_\eta \\ & + \eta_y(\mu\xi_x v_\xi)_\eta + \eta_y(\mu\eta_x v_\eta)_\eta.\end{aligned} \tag{2.2}$$

The equations for v and T are also transformed. The differencing of the above equations will be illustrated using only the first two terms on the right hand side of the above equation since the other terms are differenced in a similar manner. The first term is differenced as

$$\frac{4}{3}(\xi_x)_i\left\{(\mu\xi_x)_{i+\frac{1}{2},j}\left(\frac{u_{i+1,j}-u_{i,j}}{\Delta\xi}\right) - (\mu\xi_x)_{i-\frac{1}{2},j}\left(\frac{u_{i,j}-u_{i-1,j}}{\Delta\xi}\right)\right\}/\Delta\xi, \tag{2.3}$$

where $(\mu\xi_x)_{i+\frac{1}{2},j} = \frac{1}{2}(\mu_{i,j}(\xi_x)_i + \mu_{i+1,j}(\xi_x)_{i+1})$ and

$(\xi_x)_i = \xi_x(\xi_i)$, $\mu_{i,j} = \mu(T_{i,j})$. The second term on the right-hand side of equation (2.2) is differenced as

$$\frac{4}{3}(\eta_x)_{i,j}\left\{(\mu\xi_x)_{i,j+1}\left(\frac{u_{i+1,j+1} - u_{i-1,j+1}}{2\Delta\xi}\right)\right. \\ \left. - (\mu\xi_x)_{i,j-1}\left(\frac{u_{i+1,j-1} - u_{i-1,j-1}}{2\Delta\xi}\right)\right\}\frac{1}{2\Delta\eta} \quad . \tag{2.4}$$

Denoting the difference operator for the right-hand side of equation (2.1) by V, the algorithm for the viscous terms can be written as

$$\begin{aligned} \bar{W}_{i,j} &= W_{i,j} + \Delta t\ V_{i,j}(W), \\ W^*_{i,j} &= \tfrac{1}{2}\{W_{i,j} + \bar{W}_{i,j} + \Delta t\ V_{i,j}(\bar{W})\}, \end{aligned} \tag{2.5}$$

where

$$W = \begin{pmatrix} \rho \\ u \\ v \\ T \end{pmatrix} \quad .$$

Note that the factor of $1/3$ on the left of equation (2.1) cancels with the $1/3$ factor from the time step of $1/3\ \Delta t$. This defines the viscous splitting operator $S_V(\Delta t)$, i.e.

$$W^* = S_V(\Delta t)W. \tag{2.6}$$

The hyperbolic portion of the splitting for the two-dimensional equations

can be written as

$$\frac{2}{3} U_t + F_x + G_y = 0, \tag{2.7}$$

where

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix} , \qquad F = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E+p)u \end{pmatrix} , \qquad G = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E+p)v \end{pmatrix} .$$

Using the coordinate transformation (2.2) the equations in the (ξ,η) coordinates become

$$\frac{2}{3} U_t + \xi_x F_\xi + \eta_x F_\eta + \eta_y G_\eta = 0. \tag{2.8}$$

Equation (2.8) is split into the two one-dimensional systems

$$\frac{1}{3} U_t + \xi_x F_\xi = 0 , \tag{2.9}$$

and

$$\frac{1}{3} U_t + \eta_x F_\eta + \eta_y G_\eta = 0. \tag{2.10}$$

The difference scheme for equation (2.9) is

$$\begin{aligned} \bar{U}_{i,j} &= U_{i,j} - \frac{\Delta t}{\Delta \xi}(\xi_x)_i (F_{i+1,j} - F_{i,j}) , \\ U^*_{i,j} &= \tfrac{1}{2}\left\{ U_{i,j} + \bar{U}_{i,j} - \frac{\Delta t}{\Delta \xi}(\xi_x)_i (\bar{F}_{i,j} - \bar{F}_{i-1,j}) \right\} . \end{aligned} \tag{2.11}$$

where $\bar{F}_{i,j} = F(\bar{U}_{i,j})$. The difference scheme for equation (2.10) is

$$\bar{U}_{i,j} = U_{i,j} - \frac{2\Delta t\, a^-_{i,j}}{y_{i,j+1} - y_{i,j-1}} \left(b_{i,j}(F_{i,j} - F_{i,j-1}) + (G_{i,j} - G_{i,j-1}) \right) ,$$

$$U^*_{i,j} = \tfrac{1}{2}\left(U_{i,j} + \bar{U}_{i,j} - \frac{2\Delta t\, a^+_{i,j}}{y_{i,j+1} - y_{i,j-1}} \left(b_{i,j}(\bar{F}_{i,j+1} - \bar{F}_{i,j}) + (\bar{G}_{i,j+1} - \bar{G}_{i,j}) \right) \right) \tag{2.12}$$

where

$$\bar{F}_{i,j} = F(\bar{U}_{i,j}) \quad , \quad \bar{G}_{i,j} = G(\bar{U}_{i,j}) ,$$

and

$$b_{i,j} = \left. \frac{\eta_x}{\eta_y} \right|_{i,j} , \qquad a^+_{i,j} = 1/a^-_{i,j} = (y_{i,j} - y_{i,j-1})/(y_{i,j+1} - y_{i,j}).$$

Note that the scheme (2.12) is second-order accurate in time and space.

Equations (2.11) and (2.12) define the splitting operators $S_\xi(\Delta t)$ and $S_\eta(\Delta t)$, respectively. The total scheme is then defined by

$$U^{2n+1}_{i,j} = T^{-1} S_V(\Delta t) T \; S_\eta(\Delta t) \; S_\xi(\Delta t) \; U^{2n}_{i,j},$$

$$U^{2n+2}_{i,j} = S_\xi(\Delta t) \; S_\eta(\Delta t) \; T^{-1} \; S_V(\Delta t) T \; U^{2n+1}_{i,j}, \tag{2.13}$$

where T is the transformation from the conserved variables $U = (\rho, \rho u, \rho v, E)$ to the physical variables $W = (\rho, u, v, T)'$, i.e.

$$W = T(U).$$

Note that the operator sequences in (2.13) are reversed in successive time steps, this is done to maintain the overall second-order accuracy in time. The stability for each scheme of the splitting requires that the time step for the scheme be no larger than the largest allowable time step for each splitting. Let

$$\Delta t_\xi = \min_{i,j} \frac{\Delta\xi}{(\xi_x)_i(|u_{i,j}| + c_{i,j})} ,$$

$$\Delta t_\eta = \min_{i,j} \frac{(y_{i,j+1} - y_{i,j-1})/2}{|v_{i,j}| + |\frac{\eta_x}{\eta_y}|_{i,j} |u_{i,j}| + c_{i,j}\sqrt{1 + \left(\frac{\eta_x}{\eta_y}\right)^2_{i,j}}} ,$$

$$\Delta t_V = \min_{i,j} \frac{\sigma\rho_{i,j}}{2\gamma\mu_{i,j}} \left(\left(\frac{\xi_{x,i}}{\Delta\xi}\right)^2 + \left(\frac{\eta_{y,i,j}}{\Delta\eta}\right)^2 + \left(\frac{\eta_{x,i,j}}{\Delta\eta}\right)^2 \right)^{-1} ,$$

where $c_{i,j} = \sqrt{\gamma p_{i,j}/\rho_{i,j}}$ is the local sound speed. The time step Δt was chosen as

$$\Delta t = .9 \quad \min(\Delta t_\xi, \Delta t_\eta, \Delta t_V).$$

Because of the presence of the transformation T in the scheme (2.13) the stability of the overall scheme does not follow immediately from the stability of each of the splitting schemes. However, Abarbanel and Gottlieb (1980) show that the linearized Navier-Stokes equations can be transformed to a system which is symmetric. In this symmetric form it is easily seen that the linearization of the time-split scheme (2.13) is stable since each of the splittings is stable, (see Abarbanel and Gottlieb (1980)).

Further splitting of the parabolic operator as advocated by Abarbanel and Gottlieb would also be efficient on vector processor.

3. THE APPLICATION

A sketch of the slotted nozzle for which computations were made is given in Figure 1. In the converging portion of the nozzle the flow is subsonic and accelerating. The flow becomes supersonic in the throat region of the nozzle and continues accelerating in the diverging portion. It has been shown experimentally by Anders et al. (1977) that the turbulent boundary layer on the tunnel wall propagates disturbances which interfere with measurements in the test section. The suction slot removes the upstream boundary layer so that a new boundary layer which remains laminar to a much higher Reynolds number, begins at the slot lip. The disturbance level of the flow in the test section is then considerably reduced.

The slot is so designed that the flow within it quickly becomes supersonic, and for the computations the outflow boundary in the slot was chosen so that the flow there would be supersonic.

The computational grid for an inviscid flow is shown in Figure 2. A finer grid spacing was used near the walls to resolve the boundary layers for viscous flow computations. The computational domain consisted of only the area above the centerline since the flow was assumed to be symmetric about the centerline.

The boundary conditions for the differential equations are as follows. At inflow, the density, temperature, and flow angle were specified, i.e.

$$\rho = \rho_0, \quad T = T_0, \quad \text{and} \quad u = v \tan\theta(y), \tag{3.1}$$

where $\theta(y)$ is a specified function. Three boundary conditions are appropriate at a subsonic inflow boundary (Oliger and Sundstrom (1978)). At outflow the flow is supersonic and no quantities are specified. Along the wall of the tunnel the no-slip and adiabatic conditions were used, i.e.

$$u = v = 0 \quad \text{and} \quad \frac{\partial T}{\partial n} = 0.$$

The above inflow boundary conditions were chosen because from a one-dimensional analysis of steady nozzle flow (see Courant and Friedrichs (1948, p. 377f)) the Mach number at inflow, which is proportional to $u/\sqrt{T}$, and the mass flux at inflow, which is the integral of ρu, are determined by the conditions at the nozzle throat and the cross-sectional area ratios. Thus, if the inflow velocity component u were prescribed along with either T or ρ, then a steady flow may not have developed. Also, physically the steady flow is determined by the temperature and pressure, and hence density, of the fluid for upstream of the throat. Therefore, among all choices of three physical variables to specify at the inflow that given by (3.1) is most natural.

For the difference approximation, all the variables at the outflow boundary were determined by extrapolation from the interior. At the inflow boundary, the velocity component u was determined by extrapolation and then the component v was computed by equation (3.1). Along the wall of the tunnel the density was determined by extrapolation.

4. IMPLEMENTATION

To utilize the capabilities of the STAR to their best advantage the problem was organized so that as much of the computation as possible was done through long vector operations. The scheme presented in section 2 when applied to the problem of section 3 is quite suitable for a vector processor.

The problem had the advantage of being able to fit entirely in core memory on the STAR and thus avoided the difficulties of paging data from the virtual memory. The language used to encode the program was SL/1 which was developed at NASA Langley Research Center by Knight (1979).

To implement the difference scheme (2.9) on the STAR the values of each variable, ρ, ρu, ρv, E, etc. were assigned to a vector of length L equal to the total number of grid points. Thus, referring to Figure 3 which illustrates the computational domain, $\rho(1)$, $\rho(2),\cdots$, $\rho(IA)$, $\rho(IA+1),\cdots$ are the values of the density at the grid points a_1, $a_2,\cdots$, a_{IA}, $a_{IA+1},\cdots$. There are $LA = IA \times JA$ points in the lower region, region A, and $LB = IB \times JB$ points in the upper region, region B. The points

$$b_1,\cdots,b_{IC},$$

represent the same physical points as the points

$$a_k,\cdots, a_{k+IC-1},$$

but are distinct in the computational domain and correspond to distinct vector locations. This line of grid points which is common to both regions and is doubly represented is called the common line. Note that there is

no special differencing used at these points and the grid coordinate mapping is smooth across this line.

A vector then consisists of $L = LA + LB$ locations, one location for each point in region A and region B.

In addition to vectors whose elements take on numerical values the STAR also employs bit vectors. Bit vectors are vectors whose elements take on logical values and are used to mask out operations. Thus if the vectors A and B are to be added under the control of the bit vector b to give the result C then when the k*th* element of b is FALSE the sum of the k*th* elements of A and B is computed but is not stored in the k*th* location of C. If the k*th* element of b is TRUE then the sum is computed and stored in the k*th* location of the vector C.

Consider now the simplest splitting (2.11). Each component of the flux F is represented by a vector and is computed by vector instructions. Then the forward difference is taken for all points from index 1 to L-1 and the result is multiplied by $\xi_x \Delta t/\Delta\xi$. Each of these are vector operations of length L-1. However, for those indices k for which the $(k+1)\delta t$ grid point is not a neighbor of the k*th* grid point, the "forward" difference taken between the fluxes at the indices k and k+1 is a physically meaningless quantity. Thus when the results of the forward difference operation are added to the value of the components of U, a bit vector is employed to mask out those additions involving physically meaningless quantities. By this means the predicted quantities $\bar{U}$ are computed by equation (2.11). Those elements of $\bar{U}$ for which the results of (2.11) where masked by the bit vector are computed by different means such as extrapolation or leaving the value unchanged if it is a specified boundary value. For example, if zeroth-order extrapolation is used then the components of $\bar{U}$ at the k*th*

location are set equal to the values of $\bar{U}$ at the $(k-1)\delta t$ location under the control of the bit vector which is the negation of that mentioned above. The corrector portion of equation (2.11) is done similarly.

A computation such as the extrapolation in the example just mentioned in which the all but a small percentage of the points are masked may be considered wasteful. However, on the STAR-100A the alternative is to employ scalar operations which is quite slow. Thus these "wasteful" operations may be the fastest. On the CYBER 203, which has a much faster speed for scalar arithmetic, it is almost certainly more economical to handle the boundary conditions with scalar operations.

The splitting (2.12) is implemented in a similar fashion, but with the difference that the forward and backward differences in the η variable require three vector operations due to the geometry of the computational domain. For a forward difference, the first operation is for the lower region A and involves points whose indices in the vector differ by IA. This is because grid points in region A which are neighbors in the η direction correspond to vector locations a distance of IA apart. This operation is valid for all points of region A except the top row. The second vector operation is to do the same thing for region B, but here the neighbors occupy locations which are IB apart in the vector.

The third operation is to set the values for the points of the common line which are in part A of the vector, and not computed in the first operation, equal to the corresponding points in part B of the vector which were computed in the second operation. These vector operations are of lengths LA - IA, LB - IB, and IC, respectively. The remaining points are again treated by an extrapolation.

To implement the viscous splitting operator required a large number of temporary vectors. Therefore care was taken so that few, if any, temporary vectors were generated by the compiler to evaluate complicated expressions. Thus all computations to compute $V(W)$ in equation (2.5) were written to involve only two operands. Also the terms such as those in equation (2.4) were computed in such an order so as to minimize as much as possible the number of temporary vectors that had to be carried along in the computation. For example, the values of $(\mu\xi_x)_{i+\frac{1}{2},j}$ were computed and stored in a temporary vector and all those divided differences requiring $(\mu\xi_x)_{i+\frac{1}{2},j}$ were computed and stored in a temporary vector and all those divided differences requiring $(\mu\eta_x)_{i+\frac{1}{2},j}$ were computed and added to W. Next $(\mu\eta_x)_{i,j+\frac{1}{2}}$ was computed and stored in the same temporary vector which had stored $(\mu\eta_x)_{i+\frac{1}{2},j}$. After all divided differences requiring $(\mu\eta_x)_{i,j+\frac{1}{2}}$ were computed and added to W the temporary vector was used for another quantity. Through the use of the LITERALLY statement in SL/1 this same temporary vector could be given a different appropriate name each time its use changed. In this way the viscous splitting (2.5) could be implemented without using an excess number of temporary vectors which may have caused the storage to exceed the core storage.

As with the η-splitting most of the differences in the viscous splitting involved three vector operations, one for region A, one for region B, and one for the common line. The program had to be well documented with comments to make it understandable.

The calculations were performed using 32-bit arithmetic. The standard word size on the STAR is 64 bits but through the use of SL/1 the half-word arithmetic can be used giving a substantial increase of speed without a serious loss of accuracy.

RESULTS

The numerical method and computer code described here has been used in design studies for several nozzles, both two-dimensional and axisymmetric. We present here only the results for a particular two-dimensional slotted nozzle.

The nozzle contour with the numerically generated grid is shown in Figure 2. The total number of grid points was 12006. The stagnation pressure and temperature were 3600 psf and 520° R respectively and the Reynolds number was 3.2×10^5, being evaluated at the sonic line on the center line using the throat radius as the reference length.

The computations were started with a flow field derived from an inviscid one-dimensional analysis (see e.g. Courant and Friedrichs (1948, p. 377f)) with a correction for a boundary layer. The computations were stopped when the maximum relative change in the density between two successive time steps was sufficiently small or when the solution appeared to be nearly converged in the main portion of the nozzle. For the results discussed below this was achieved in 40,000 time steps in several stages of 2000 or 8000 steps each. The cpu time per time step per grid point was about 1.1×10^{-5} seconds on the CYBER-203.

The Mach number contours for this solution are shown in Figure 4. The Mach number varied from about .01 at the inflow to 2.0 at the main outflow and 1.7 at the slot outflow. Figure 5 shows the stream function computed for this run. The stream function was computed as an approximation to

$$\psi = \int_0^y \rho u \, dy.$$

Further results and discussion of these results are presented in Beckwith and Holley (to appear).

APPENDIX

Numerical Grid Generation

The procedure for numerically computing the grid for the slotted nozzle calculations did not use the vector processor, but for completeness it will be briefly described.

The numerical method for the grid generation is essentially that of Thompson, et al. (1974). The basic idea is to map the computational domain consisting of rectangles joined along a portion of one side onto the physical domain (see Fig. 3). The mapping function $x(\xi)$ was specified as a piecewise polynomial defined so that $x'(\xi)$ was small where clustering of grid lines was desired, and $x''(\xi)$ was continuous. The function $y(\xi,\eta)$ was determined requiring that the inverse function $\eta(x,y)$ satisfy

$$\frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2} = f(\xi,\eta), \tag{A.1}$$

on the region surrounded by the contour with η specified on the boundaries. The function $f(\xi,\eta)$ will be described later.

If the Jacobian J of the mapping is nonvanishing then equation (A.1) can be transformed to

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} + J^2(\xi_{xx} y_{\xi} + f(\xi,\eta) y_{\eta}) = 0, \tag{A.2}$$

on the computational domain illustrated in Figure 2. The coefficients in equation (A.2) are given by

$$\alpha = y_{\eta}^2, \quad \beta = y_{\xi} y_{\eta}, \quad \gamma = x_{\xi}^2 + y_{\xi}^2, \quad \text{and} \quad J = x_{\xi} y_{\eta},$$

and note that $\xi_{xx} = -x_{\xi\xi}/(x_{\xi})^3$.

The values of y were specified on the boundary and thus equation (A.2) is an nonlinear elliptic boundary value problem with Dirichlet boundary data.

The equation (A.2) was solved by a iterative procedure similar to SOR but with a variable iteration parameter. The variable parameter was required because of the high degree of stretching in the grid transformation. The procedure is discussed in Strikwerda (1980).

The function $f(\xi,\eta)$ was determined as follows. On the boundaries on which x is constant, $\xi = \xi_0$ and f was specified by

$$f(\xi_0,\eta) = \eta_{yy} = -y_{\eta\eta}/(y_\eta)^3. \tag{A.3}$$

This formula is obtained by observing that at these boundaries it is desirable to have the coordinate lines to be nearly straight, i. e. $\eta_{xx} = 0$. If $\eta_{xx} = 0$ then from (A.1) we obtain (A.3). Along the boundaries $\eta = 0$ and $\eta = 1$, the value of f was set to zero. To cluster the grid points near the lip of the slot a large positive value was assigned to f at the grid point at the lip. The function $f(\xi,\eta)$ was then determined by a polynomial interpolation from these boundary points.

By adjusting the value of f at the tip, the polynomial interpolation, and the relative number of grid points in regions A and B, satisfactory grids were obtained quite easily.

ACKNOWLEDGMENT

The author would like to acknowledge the programming assistance of Geoffrey Tennille and the technical assistance of Ivan Beckwith.

REFERENCES

Strang, G. (1968)

On the construction and comparison of difference schemes.

SIAM J. Numer. Anal., 5, p. 506-517.

Gottlieb, D. (1972)

Strang type difference schemes for multidimensional problems.

SIAM J. Numer. Anal., 9, p. 650-661.

Abarbanel, S. and Gottlieb, D. (1980)

Optimal time splitting for two and three dimensional Navier-Stokes equations with mixed derivatives

ICASE Report No. 80-6.

Strikwerda, J. (1980)

Iterative methods for the numerical solution of second order elliptic equations with large first order terms.

SIAM J. on Sci. and Stat. Comp., 1, p. 119-130.

Thompson, J. F., Thames, F. C., and Mastin, C. W. (1974)

Automatic numerical generation of body-fitted curvilinear coordinate system for fields containing any number of arbitrary two-dimensional bodies.

J. Computational Phys., 15, p. 299-319.

Walsh, J. and Strikwerda, J. (1980)

Numerical solutions of compressible Navier-Stokes equations for two-dimensional, laminar, hypersonic flow at a wing-elevon junction.

(to appear)

Oliger, J. and Sundstrom, A. (1978)

Theoretical and practical aspects of some initial boundary value problems in fluid dynamics.

SIAM J. Appl. Math., 35, p. 419-446.

Thomas, P. D. (1979)

Numerical method for predicting flow characteristics and performance of nonaxisymmetric nozzles-theory.

NASA Contractor Report 3147.

Rae, W. J. (1971)

Some numerical results on viscous low-density nozzle flows in the slender-channel approximation.

AIAA J., 9, p. 811-820.

Mitra, N. K. and Fiebig, M. (1975)

Low Reynolds number hypersonic nozzle flow.

Z. Flugwiss, 23, p. 39-45.

MacCormack, R. W. (1971)

Numerical solution of the interaction of a shock wave with a laminar boundary layer.

Lecture Notes in Physics, 8, Springer-Verlag, p. 151.

Anders, J. B., Stainback, P. C., Keefe, L. R., and Beckwith, I. E. (1977)

Fluctuating disturbances in a Mach 5 wind tunnel

AIAA J., 15, p. 1123-1129.

Beckwith, I. E. (1975)

Development of a high Reynolds number quiet tunnel for transition research.

AIAA J., 13, p. 300-306.

Cline, M. C. (1976)

Computation of two-dimensional, visocus nozzle flow.

AIAA J., 14, p. 295-296.

Beckwith, I. E. and Holley, B. B. (to appear)

The design of rapid expansion nozzles for quiet wind tunnels with preliminary test data at Mach numbers 3 and 3.5.

Courant, R. and Friedrichs, K. O. (1948)

Supersonic Flow and Shock Waves.

Interscience Publishers, Inc., New York, NY.

Knight, J. C. (1979)

SL/1 Reference Manual,

Analysis and Computation Division, NASA Langley Research Center, Hampton, VA 23665.

Figure Captions

Figure 1. A diagram of a two-dimensional nozzle with suction slots.

Figure 2. Grid lines for the slotted nozzle calculation. Insert shows slot region enlarged.

Figure 3. A schematic diagram of the computational grid and mapping discussed in Section 4 and the appendix.

Figure 4. Contour plot of Mach number.

Figure 5. Contour plot of the streamfunction.

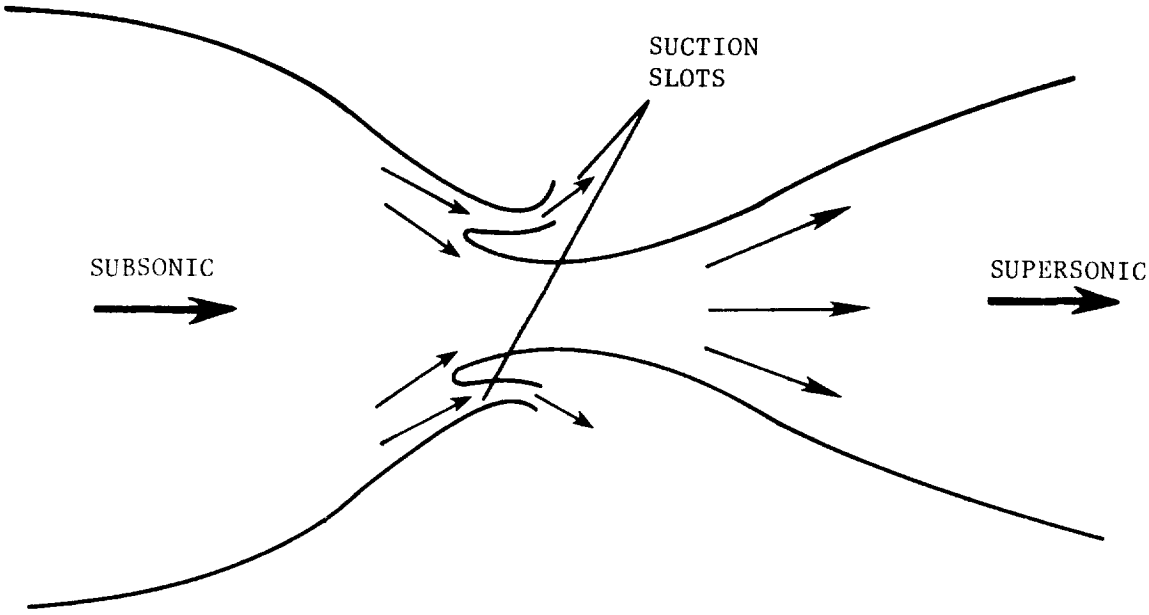

Figure 1.

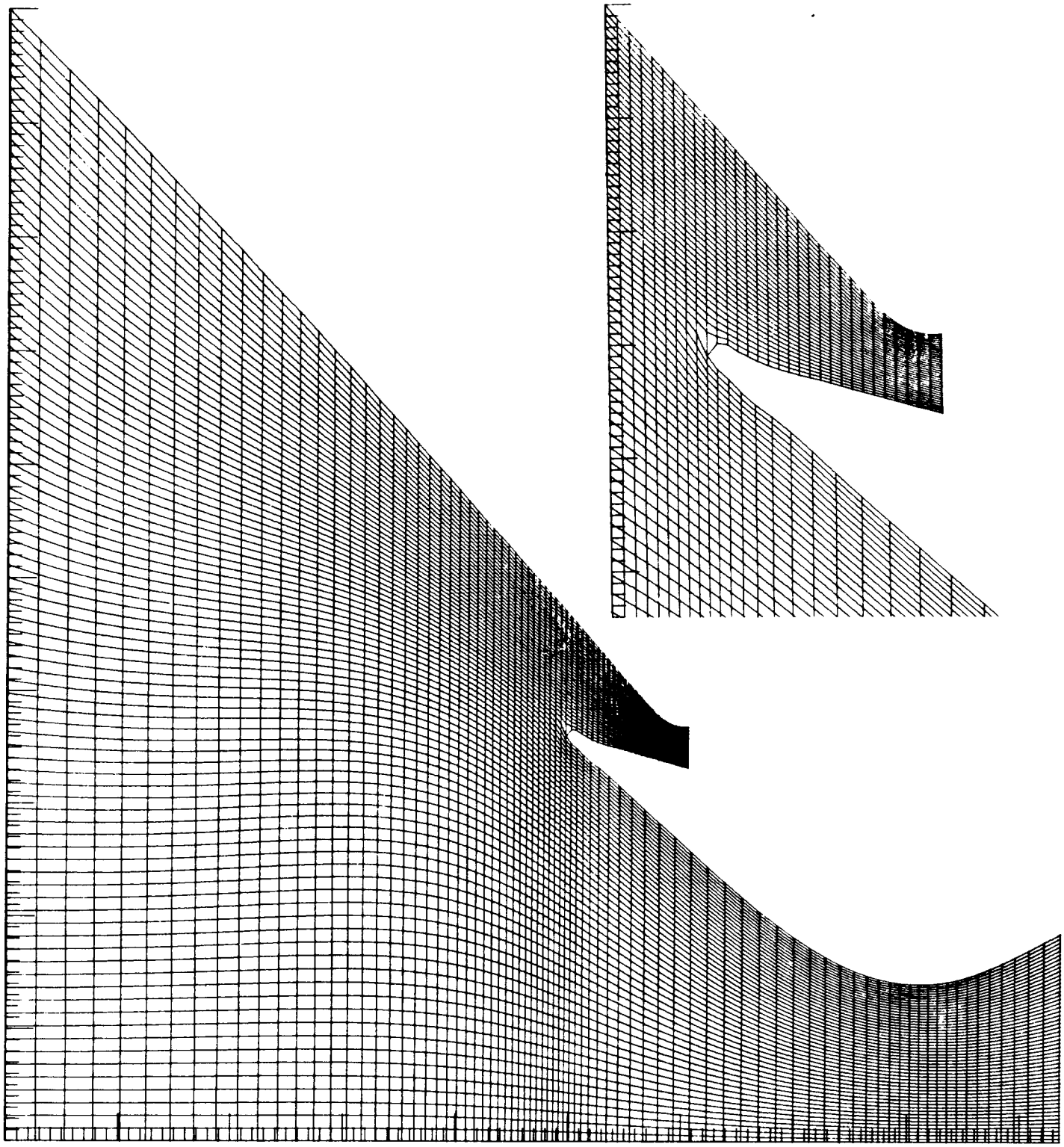

Figure 2.

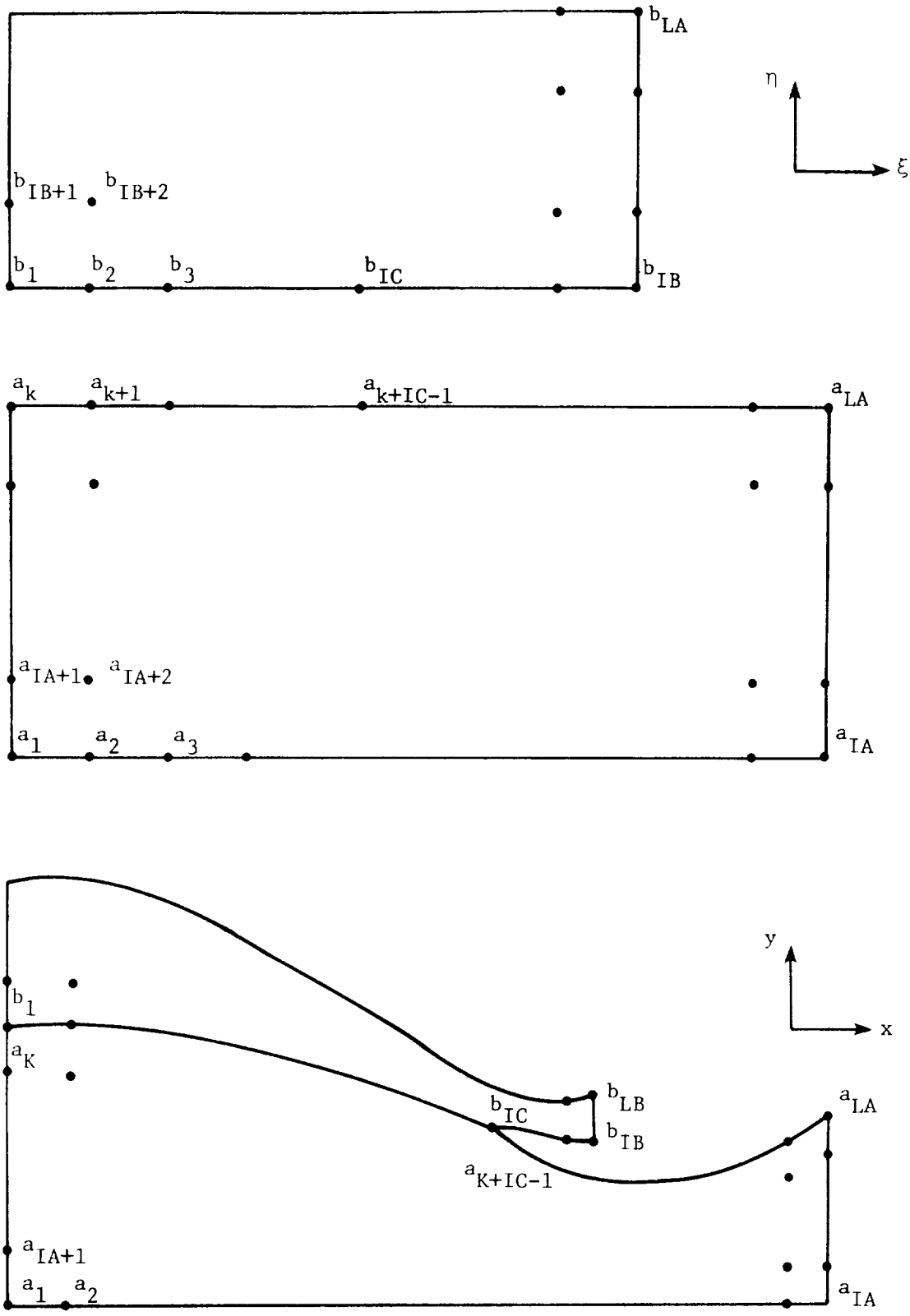
b_{LA}
η
ξ
b_{IB+1}
b_{IB+2}
b_1
b_2
b_3
b_{IC}
b_{IB}
a_k
a_{k+1}
a_{k+IC-1}
a_{LA}
a_{IA+1}
a_{IA+2}
a_1
a_2
a_3
a_{IA}
y
x
b_1
a_K
b_{IC}
b_{LB}
b_{IB}
a_{K+IC-1}
a_{LA}
a_{IA+1}
a_1
a_2
a_{IA}

Figure 3.

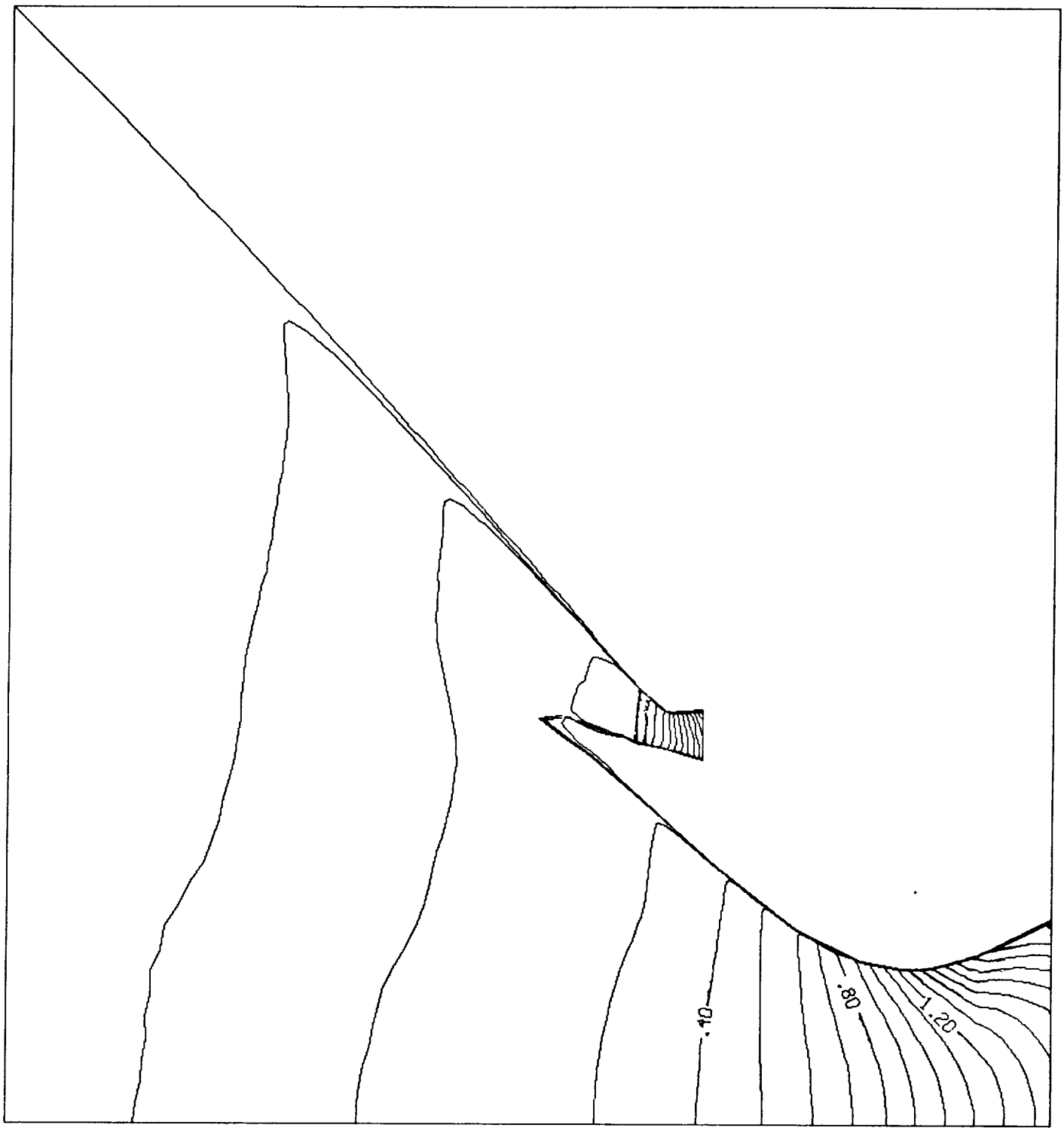

Figure 4.

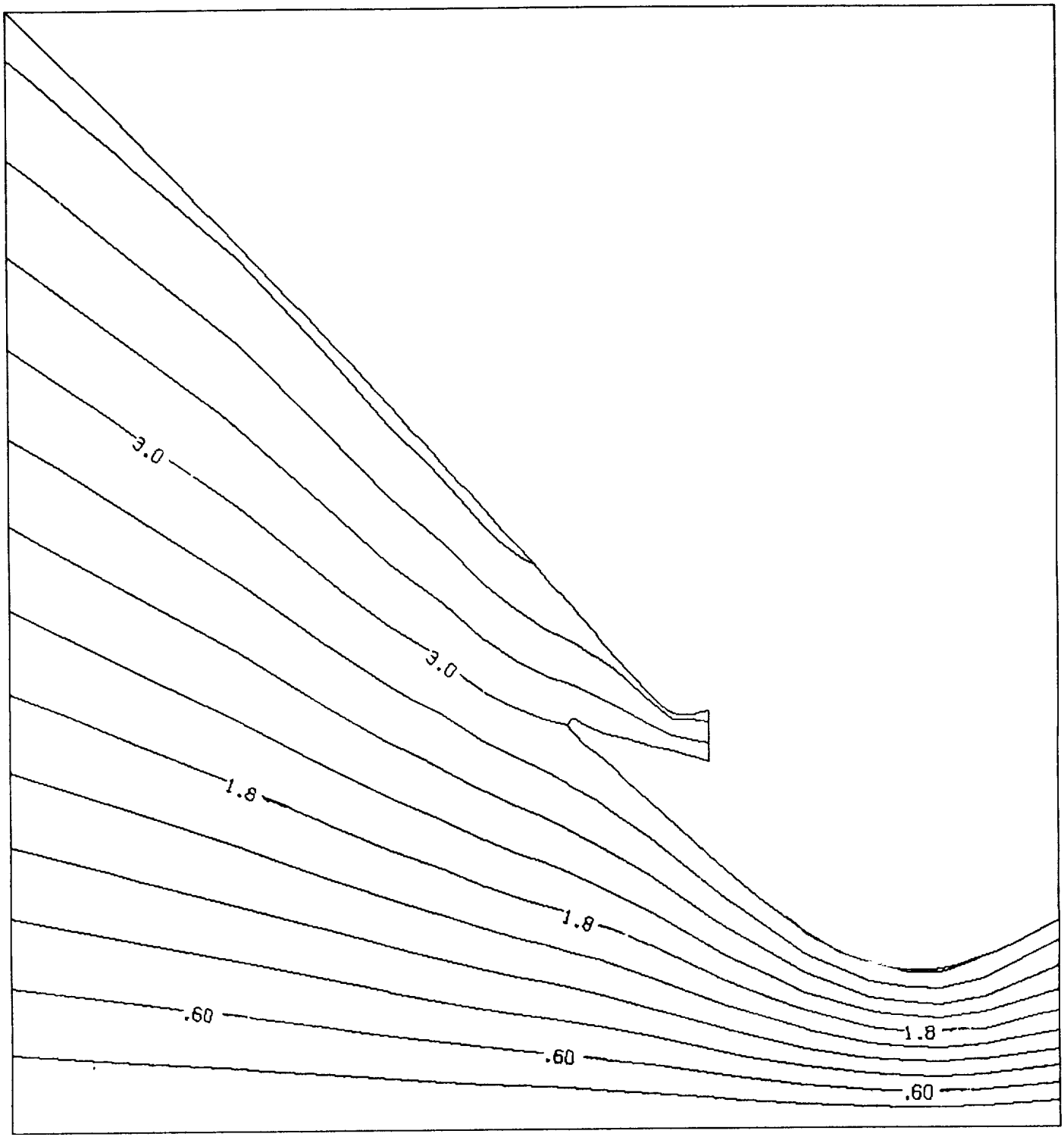

Figure 5.